Advanced Dynamical Systems (MATH60146/70146)

Coursework 4

Homework issued on: February 14, 2025
Due date: February 21, 2025

Duration: 1 week | Spring 2024-25 | Max mark: 15 Points

Objective: This coursework focuses on computing Lyapunov exponents and bifurcation diagrams for the Hénon map and exploring fractal structures in nature.

1. **Lyapunov Exponent and Bifurcation Diagram for the Hénon Map** (15)

   The **Hénon map** is a discrete-time dynamical system given by the recurrence equations:

   $$x_{n+1} = 1 - ax_n^2 + y_n, \quad y_{n+1} = bx_n.$$

   where $(x_n, y_n) \in \mathbb{R}^2$ define the state space of the system, and $(a, b) \in \mathbb{R}^2$ are system parameters. The map was introduced by Michel Hénon as a simplified model of the Poincaré section of a dynamical system describing a two-dimensional dissipative system.

   For this coursework, we consider the classical Hénon parameters:

   $$a \in [0.2, 1.4], \quad b = 0.3.$$

   The initial condition should be chosen within the attractor basin, e.g., $(x_0, y_0) = (0, 0)$.

   (a) Compute both **Lyapunov exponents** λ_1, λ_2 for the Hénon map over the given range $a \in [0.2, 1.4]$, keeping $b = 0.3$ fixed. Use the full **Jacobian matrix** and apply **Gram-Schmidt orthonormalization** to ensure numerical accuracy. **Computation details:** Iterate the system for **5000 steps**, discarding the first **1000 steps** to remove transients. The remaining **4000 iterations** should be used to compute the Lyapunov exponents.

   (b) Generate the **bifurcation diagram** by plotting the long-term values of x_n as a function of a. **Computation details:** Iterate the system for **5000 steps**, discarding the first **1000 steps** to remove transients. The remaining **4000 points** should be plotted for each a to visualize the attractor structure.

   (c) Visualize the results in a **bottom-up layout**:

   - The upper subplot should display the **bifurcation diagram**.
   - The lower subplot should contain both **Lyapunov exponents** λ_1 **and** λ_2 on the same plot, clearly distinguishing them.

   Mark the zero-line on the Lyapunov exponent plot to highlight bifurcation points.

   Note: Due to numerical sensitivity, Python may generate occasional warnings for certain values of a during the Lyapunov exponent computation. These warnings can be ignored by suppressing them in Python using:

   ```
   import warnings
   warnings.filterwarnings("ignore")
   ```

2. ***Bonus Question: Fractal Dimension of a Leafless Tree**

   Fractals appear in natural patterns such as tree branches, coastlines, and clouds. In this question, you will compute the **fractal dimension** of a leafless tree using the *box-counting method*.

(a) Obtain an image of a **tree with no leaves** (bare branches) that exhibits fractal-like structure. If taking a picture is not feasible, find a suitable image online.

(b) Convert the image into a **binary matrix** where pixels corresponding to branches are marked as 1 and the background as 0. Use Python's 'cv2' library to process the image. **Installation:** If 'cv2' (OpenCV) is not installed, you can install it using:

```
pip install opencv-python
```

(c) Since images are in pixels but fractal structures exist in real-world scales, we must **approximate the real-world frame width**:

- Assume the tree width is about **10 meters** (or adjust based on known scale).
- Convert **pixel box sizes** into **real-world scales** using:

$$\text{real\_box\_size} = \left(\frac{\text{box\_size (pixels)}}{\text{image\_width (pixels)}} \right) \times \text{frame\_width}$$

- This ensures the computed fractal dimension is **scale-invariant**.

(d) Compute the **fractal dimension** using the *box-counting method* for box sizes in pixels: [4, 8, 16, 32, 64, 128, 256]. The resolution of the image can be 1000 $\times$ 1000. If the figure is not square, then the resolution can be optimized.

(e) Plot:

- The binary representation of the tree (black = tree, white = background).
- A log-log plot of the box-counting method, showing the relationship between $\log(1/\varepsilon)$ and $\log(N(\varepsilon))$.

(f) Estimate and report the fractal dimension D of the tree.

Note that: If you are not sure about any computation details, please use the same parameters or techniques as in the lecture notes. You do not need to be very precise with the details for the bonus questions. Show your creativity!

*Bonus questions are designed as an optional challenge for students who wish to explore deeper insights into complexity. While they are primarily for enrichment, they can also contribute to the final grade in specific cases. If a student's total score is very close to the next grade boundary (within 1-2 points), completing the bonus question may help bridge the gap. However, bonus questions cannot contribute more than 2 additional points in total.

Submission Guidelines: Submit a single PDF document containing your code, solutions, justifications, and figures.

Best wishes,